



# Optimized Lifting Scheme Based on a Dynamical Fully Connected Network for Image Coding

Tassnim Dardouri, Mounir Kaaniche, Amel Benazza-Benyahia,  
Jean-Christophe Pesquet

## ► To cite this version:

Tassnim Dardouri, Mounir Kaaniche, Amel Benazza-Benyahia, Jean-Christophe Pesquet. Optimized Lifting Scheme Based on a Dynamical Fully Connected Network for Image Coding. ICIP 2020 - The 27th IEEE International Conference on Image Processing, Oct 2020, Virtual, United Arab Emirates. 10.1109/ICIP40778.2020.9191292 . hal-03139684

**HAL Id: hal-03139684**

**<https://hal.archives-ouvertes.fr/hal-03139684>**

Submitted on 12 Feb 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# OPTIMIZED LIFTING SCHEME BASED ON A DYNAMICAL FULLY CONNECTED NETWORK FOR IMAGE CODING

*T. Dardouri<sup>1</sup>, M. Kaaniche<sup>1</sup>, A. Benazza-Benyahia<sup>2</sup>, and J.-C. Pesquer<sup>3</sup>*

<sup>1</sup> Université Sorbonne Paris Nord,  
L2TI, UR 3043, F-93430,  
Villetaneuse, France  
tasnim.dardouri@edu.univ-paris13.fr  
mounir.kaaniche@univ-paris13.fr

<sup>2</sup> University of Carthage, SUP'COM,  
LR11TIC01, COSIM Lab,  
Cité Technologique des Communications,  
2083 Ariana, Tunisia  
benazza.amel@supcom.rnu.tn

<sup>3</sup> Université Paris Saclay,  
CentraleSupélec,  
Centre de Vision Numérique, Inria,  
91192 Gif-sur-Yvette, France  
jean-christophe@pesquet.eu

## ABSTRACT

Wavelet decompositions based on lifting schemes have been widely used in image coding. Generally, the efficiency of such compression methods strongly depends on the design of the lifting operators, namely the prediction and update filters. To improve their performance, we propose in this paper to optimize these filters by resorting to two learning strategies. In the first one, classical Fully Connected Networks (FCNs) are exploited to perform the prediction and update. In the second approach, we develop an adaptive learning method that takes into account the input image, yielding a dynamical model of FCN. Experimental results, carried out on the standard Challenge Learned Image Compression (CLIC) dataset, show the benefits that can be drawn from the proposed approaches compared to conventional ones.

**Index Terms**— Lifting schemes, image compression, adaptive wavelets, optimization, neural networks.

## 1. INTRODUCTION

For several decades, wavelets have attracted much attention in various signal and image processing fields like compressed sensing, super-resolution, and compression [1, 2]. To generate a wavelet representation of a given signal (or image), filter banks and lifting scheme (LS) based architectures have been widely used in the literature. For instance, a great attention has been paid to LS-based decomposition because of their many advantages in terms of computational cost and their ability to guarantee perfect reconstruction of the original signal [3, 4]. For these reasons, LS has been retained in the JPEG2000 compression standard [5]. Moreover, it has been extensively employed for data compression, and more specifically, for still and stereo image coding [6] as well as video coding [7, 8]. A typical lifting structure is composed of prediction and update filters that respectively generate the detail and approximation wavelet coefficients [9]. The performance of the LS strongly depends on the choice of these filters. For this reason, various research works have been devoted to the optimization of the prediction and update operators in order to build compact wavelet representation which are well adapted to the input data contents [10, 11, 12, 13]. To this end, the prediction filters are often optimized by minimizing the  $\ell_2$ -norm of the detail coefficients [14]. The minimization of the  $\ell_1$ -norm has also been investigated in [15]. However, the optimization of the update filter is less obvious than the prediction one, and two main approaches have been developed for that purpose. The first one consists in minimizing the reconstruction error [14, 16]. The second one aims at minimizing the error between the approximation coefficients

and the output of an ideal low-pass filter [13].

In order to further improve the performance of lifting-based image coding schemes, it has been recently proposed to perform the prediction stage in the conventional separable LS by making use of a convolutional neural network [17]. However, in the latter work, a mean filter has been substituted for the update stage. More generally, other recent end-to-end compression methods have been developed using various neural network architectures and a rate-distortion loss function [18, 19, 20]. It should be emphasized that these methods do not generate integer wavelet coefficients, and so, they are not suitable for lossless compression.

In this paper, we propose to investigate neural network architectures for the design of adaptive LS-based coding techniques. To this respect, a fully connected network will be first employed to perform the prediction as well as the update in a 2D non separable lifting structure. Then, the classical FCN architecture will be improved by taking into account the input image adaptatively, yielding a *dynamical* FCN model. To this end, we propose an iterative alternating optimization algorithm to learn the parameters of this new model.

The remainder of this paper is organized as follows. In Section 2, we recall the principles of the conventional 2D LS and the optimization methodology. In Section 3, the proposed neural network-based LS is described. Finally, experimental results are presented in Section 4 and some conclusions are given in Section 5.

## 2. CONVENTIONAL 2D LIFTING SCHEME

### 2.1. Non separable lifting structure

To perform efficient decompositions of images, possibly having neither horizontal nor vertical contours, a 2D Non Separable Lifting Scheme (NSLS) composed of three prediction steps followed by an update step has been proposed and widely investigated in the literature [21, 22]. The principle of this architecture, shown in Fig. 1 is described next.

Let  $x$  and  $x_j$  respectively denote the input image and its approximation subband at resolution level  $j$  (where  $x_0 = x$ ). The considered 2D decomposition consists first in splitting the input signal  $x_j(m, n)$  into four polyphase components denoted by  $x_{0,j}(m, n) = x_j(2m, 2n)$ ,  $x_{1,j}(m, n) = x_j(2m, 2n+1)$ ,  $x_{2,j}(m, n) = x_j(2m+1, 2n)$ , and  $x_{3,j}(m, n) = x_j(2m+1, 2n+1)$ . Then, three prediction filters  $\mathbf{P}_j^{(HH)}$ ,  $\mathbf{P}_j^{(LH)}$ , and  $\mathbf{P}_j^{(HL)}$  are applied to generate three detail subbands oriented diagonally  $x_{j+1}^{(HH)}$ , vertically  $x_{j+1}^{(LH)}$  and horizontally  $x_{j+1}^{(HL)}$ , respectively. Finally, an update filter  $\mathbf{U}_j$  is used to produce the approximation subband  $x_{j+1}$ . Therefore, the

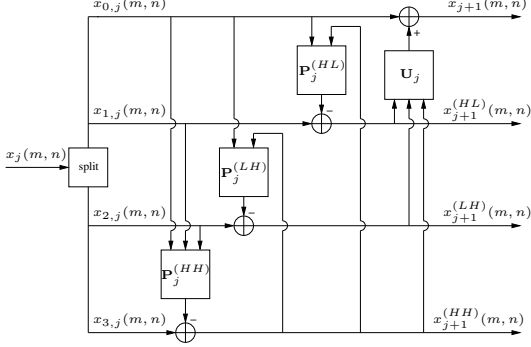


Fig. 1. NSLS decomposition structure.

resulting wavelet coefficients are as expressed as follows:

$$x_{j+1}^{(HH)}(m, n) = x_{3,j}(m, n) - \left( (\mathbf{P}_{0,j}^{(HH)})^\top \mathbf{x}_{0,j}^{(HH)}(m, n) + (\mathbf{P}_{1,j}^{(HH)})^\top \mathbf{x}_{1,j}^{(HH)}(m, n) + (\mathbf{P}_{2,j}^{(HH)})^\top \mathbf{x}_{2,j}^{(HH)}(m, n) \right), \quad (1)$$

$$x_{j+1}^{(LH)}(m, n) = x_{2,j}(m, n) - \left( (\mathbf{P}_{0,j}^{(LH)})^\top \mathbf{x}_{0,j}^{(LH)}(m, n) + (\mathbf{P}_{1,j}^{(LH)})^\top \mathbf{x}_{j+1}^{(HH)}(m, n) \right), \quad (2)$$

$$x_{j+1}^{(HL)}(m, n) = x_{1,j}(m, n) - \left( (\mathbf{P}_{0,j}^{(HL)})^\top \mathbf{x}_{0,j}^{(HL)}(m, n) + (\mathbf{P}_{1,j}^{(HL)})^\top \mathbf{x}_{j+1}^{(HH)}(m, n) \right), \quad (3)$$

$$x_{j+1}(m, n) = x_{0,j}(m, n) + \left( (\mathbf{U}_{0,j}^{(HL)})^\top \mathbf{x}_{j+1}^{(HL)}(m, n) + (\mathbf{U}_{1,j}^{(LH)})^\top \mathbf{x}_{j+1}^{(LH)}(m, n) + (\mathbf{U}_{2,j}^{(HH)})^\top \mathbf{x}_{j+1}^{(HH)}(m, n) \right),$$

where for every  $i \in \{0, 1, 2\}$  and  $o \in \{HL, LH, HH\}$ ,

- $\mathbf{P}_{i,j}^{(o)} = (p_{i,j}^{(o)}(s, t))_{(s,t) \in \mathcal{P}_{i,j}^{(o)}}$  and  $\mathbf{U}_{i,j}^{(o)} = (u_{i,j}^{(o)}(s, t))_{(s,t) \in \mathcal{U}_{i,j}^{(o)}}$  are the prediction and update weight vectors whose supports are respectively denoted by  $\mathcal{P}_{i,j}^{(o)}$  and  $\mathcal{U}_{i,j}^{(o)}$ .

- $\mathbf{x}_{i,j}^{(o)}(m, n) = (x_{i,j}^{(o)}(m + s, n + t))_{(s,t) \in \mathcal{P}_{i,j}^{(o)}}$  is a reference vector used to generate  $x_{j+1}^{(o)}(m, n)$ .

- $\mathbf{x}_{j+1}^{(HH)}(m, n) = (x_{j+1}^{(HH)}(m + s, n + t))_{(s,t) \in \mathcal{P}_{1,j}^{(LH)}}$  and  $\mathbf{x}_{j+1}^{(LH)}(m, n) = (x_{j+1}^{(LH)}(m + s, n + t))_{(s,t) \in \mathcal{P}_{1,j}^{(HL)}}$  are two reference vectors (possibly having different sizes) used to compute  $x_{j+1}^{(LH)}(m, n)$  and  $x_{j+1}^{(HL)}(m, n)$ , respectively.

- $\mathbf{x}_{j+1}^{(o)}(m, n) = (x_{j+1}^{(o)}(m + s, n + t))_{(s,t) \in \mathcal{U}_{i,j}^{(o)}}$  is the reference vector containing the set of detail samples used in the update step.

## 2.2. Prediction and update filters optimization

One key step in the design of LS-based image compression is the choice of the predictor and update operators. While the latter are kept fixed in the JPEG2000 compression standard [23], it is more interesting to optimize such operators in order to build adaptive lifting-based coding schemes which are better adapted to the image contents

[11, 13, 14]. To this respect, the prediction filters are generally optimized by minimizing the variance of the detail coefficients in the orientation  $o$ :

$$\mathcal{J}(\mathbf{P}_j^{(o)}) = \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \left( x_{i,j}(m, n) - (\mathbf{P}_j^{(o)})^\top \tilde{\mathbf{x}}_j^{(o)}(m, n) \right)^2 \quad (4)$$

where  $M_j \times N_j$  represents the number of samples  $x_{i,j}(m, n)$  to be predicted,  $\mathbf{P}_j^{(o)}$  is the prediction filter to be optimized and  $\tilde{\mathbf{x}}_j^{(o)}$  is the reference vector gathering all the coefficients used to generate the detail coefficients  $x_{j+1}^{(o)}$ . Regarding the update filter, one recent state-of-the-art method [13] aims at optimizing the update filter by minimizing the quadratic error between the approximation signal and the decimated version of the output of an ideal low-pass filter. This criterion is expressed as follows:

$$\begin{aligned} \tilde{\mathcal{J}}(\mathbf{U}_j) &= \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \left( x_{j+1}(m, n) - y_{j+1}(m, n) \right)^2 \\ &= \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \left( x_{0,j}(m, n) + \mathbf{U}_j^\top \tilde{\mathbf{x}}_{j+1}(m, n) - y_{j+1}(m, n) \right)^2 \end{aligned} \quad (5)$$

where  $\mathbf{U}_j$  is the update vector to be optimized,  $\tilde{\mathbf{x}}_{j+1}(m, n)$  is the update reference vector composed of the detail signals  $x_{j+1}^{(o)}$ , and  $y_{j+1}(m, n) = (h \star x_j)(2m, 2n)$ ,  $h$  being the impulse response of a 2D ideal rectangular low-pass filter. By minimizing the above criteria  $\mathcal{J}$  and  $\tilde{\mathcal{J}}$ , it is easy to show that the optimal prediction and update filters can be obtained by solving a linear system of equations [13].

## 3. PROPOSED FULLY CONNECTED NETWORKS BASED LIFTING SCHEME

In order to benefit from the capabilities of neural networks to perform powerful nonlinear approximation, we propose to design the prediction and update operators based on the two following FCN models.

### 3.1. Classical FCN-based LS

#### 3.1.1. FCN-based prediction stage

As the detail signal  $x_{j+1}^{(o)}$  corresponds to a prediction error, the first three prediction steps in the proposed FCN-based LS can be rewritten as

$$\begin{aligned} x_{j+1}^{(o)}(m, n) &= x_{i,j}(m, n) - \hat{x}_{i,j}(m, n) \\ &= x_{i,j}(m, n) - f_j^{(o)}(\tilde{\mathbf{x}}_j^{(o)}(m, n)) \end{aligned} \quad (6)$$

where  $x_{i,j}(m, n)$  (with  $i \in \{1, 2, 3\}$ ) is the polyphase component to be predicted and  $\hat{x}_{i,j}(m, n) = f_j^{(o)}(\tilde{\mathbf{x}}_j^{(o)}(m, n))$  is the predicted value. This value corresponds to the output of a standard FCN architecture applied to the input vector  $\tilde{\mathbf{x}}_j^{(o)}(m, n)$  which represents the reference vector used in the prediction step for generating the associated detail signal  $x_{j+1}^{(o)}(m, n)$ . More precisely, as illustrated by Fig. 2,  $L$  hidden layers are firstly used to generate the predicted value  $\hat{x}_{i,j}(m, n)$  from the input reference vector  $\tilde{\mathbf{x}}_j^{(o)}(m, n)$ . The output values of their neurons (or units) are computed by applying a linear combination (with bias) followed by a non-linear activation function. The involved parameters in these layers will be denoted by

$\Theta_j^{(o)}$ . Then, an output layer with a single neuron is applied allowing us to compute  $\hat{x}_{i,j}(m, n)$  from the unit values of the last hidden layer based on a linear combination operation. The weight parameters of this last layer will be designated by  $\mathbf{w}_j^{(o)}$ .

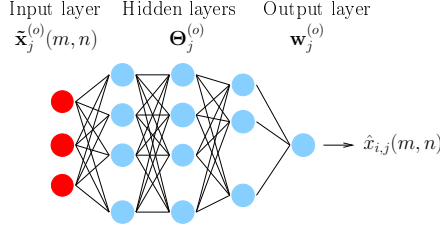


Fig. 2. FCN-based prediction stage.

The set of weights  $(\Theta_j^{(o)}, \mathbf{w}_j^{(o)})$  is learnt by performing forward and backward propagation passes, while minimizing a loss function. To this end, each FCN-based prediction stage is trained by minimizing the squared  $\ell_2$ -norm of the prediction error (i.e. the Mean Square Error (MSE)) given by

$$\mathcal{L}(x_{i,j}, \hat{x}_{i,j}) = \frac{1}{M_j \times N_j} \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} (x_{i,j}(m, n) - \hat{x}_{i,j}(m, n))^2. \quad (7)$$

To optimize the above loss function, a Mini-Batch Gradient Descent (MBGD) algorithm is employed [24]. After the convergence of the algorithm, the resulting optimal weights  $(\Theta_j^{(o)}, \mathbf{w}_j^{(o)})$  are used to compute  $\hat{x}_{i,j}(m, n)$  and then the detail wavelet coefficients  $x_{j+1}^{(o)}(m, n)$  are derived using (6). Note that this process can be separately performed on each prediction stage of the NSLS.

### 3.1.2. FCN-based update stage

It should be noted that the computation of the approximation coefficients is quite different from that of the detail coefficients, and so, it cannot be directly performed by following the previously described FCN approach. For this reason, we propose to focus on the optimization method often used to optimize the update filter (and described in Section 2). Thus, inspired from (5), we define the following error signal  $e_j(m, n)$ :

$$\begin{aligned} e_j(m, n) &= t_j(m, n) - \hat{t}_j(m, n) \\ &= t_j(m, n) - f_j^{(LL)}(\tilde{\mathbf{x}}_{j+1}(m, n)) \end{aligned} \quad (8)$$

where  $t_j(m, n) = y_{j+1}(m, n) - x_{0,j}(m, n)$ . According to (8), the update optimization problem is reformulated as a prediction problem that consists in optimally predicting the target input signal  $t_j(m, n)$  from the reference signal  $\tilde{\mathbf{x}}_{j+1}(m, n)$ . Therefore, similarly to the previous FCN-based prediction optimization tasks, the FCN-based update aims at finding the predicted value  $\hat{t}_j(m, n) = f_j^{(LL)}(\tilde{\mathbf{x}}_{j+1}(m, n))$  from the input reference vector  $\tilde{\mathbf{x}}_{j+1}(m, n)$ . The associated FCN weight parameters  $(\Theta_j^{(LL)}, \mathbf{w}_j^{(LL)})$  are optimized by minimizing a loss function similar to the previous one, i.e.  $\mathcal{L}(t_j, \hat{t}_j)$ . Once the network is trained, the resulting optimal weights allow us to compute the output  $\hat{t}_j(m, n)$  from the input reference vector  $\tilde{\mathbf{x}}_{j+1}(m, n)$ . Then, the final approximation coefficients resulting from the FCN-based update stage are deduced as follows:

$$x_{j+1}(m, n) = x_{0,j}(m, n) + \hat{t}_j(m, n). \quad (9)$$

### 3.2. Dynamical FCN-based LS

In order to design a FCN-based LS which is better *adapted* to the image to be encoded, we introduce a novel architecture, referred to as *dynamical FCN*. For this purpose, let us recall the main operations performed inside the classical FCN model, which is the composition of two main blocks. The first one, parameterized by  $\Theta_j^{(o)}$ , allows the outputs of the neurons of the hidden layers to be computed. The second block, parameterized by  $\mathbf{w}_j^{(o)}$ , produces the output of the FCN based on a linear combination of the neuron values delivered at the last hidden layer. It can thus be viewed as some kind of optimized average pooling. Subsequently, we propose to adjust the computation of the weight parameters of the second block by making them dependent on the input image. As a result, the weights of the last layer of our dynamic model depend on the input image  $x_{j,k}$ , and so, they will be denoted by  $\mathbf{w}_{j,k}^{(o)}$ , where index  $k$  indicates the  $k$ -th image  $x_{j,k}$  in the dataset. To this end, the training process of the traditional FCN model needs to be modified. More precisely, due to the dependence between the parameters  $\mathbf{w}_{j,k}^{(o)}$  and  $\Theta_j^{(o)}$ , we propose to learn this set of parameters by using an iterative algorithm that alternates between the computation of  $\mathbf{w}_{j,k}^{(o)}$  and the update of  $\Theta_j^{(o)}$ . Thus, the training process will be performed as follows. For each image  $x_{j,k}$ , two principal steps will be performed in an iterative way. Let us start from a given initialization of  $\Theta_j^{(o)}$ . In the first step of each iteration  $\ell$ , we first produce the output of the last hidden layer  $\mathbf{y}_{j,k}^{(o)}$  using the conventional FCN model. Then, the weight parameters of the last layer  $\mathbf{w}_{j,k}^{(o)}$  are computed by minimizing the  $\ell_2$ -norm of the prediction error. This minimization results in the following weights:

$$\mathbf{w}_{j,k}^{(o,\ell)} = \left( \mathbb{E}[\mathbf{y}_{j,k}^{(o)}(m, n) \mathbf{y}_{j,k}^{(o)}(m, n)^\top] \right)^{-1} \mathbb{E}[x_{i,j,k}(m, n) \mathbf{y}_{j,k}^{(o)}(m, n)]. \quad (10)$$

After that, in the second step, we update the weights of the hidden layers while setting the weights of the output layer to the values obtained in the first step  $\mathbf{w}_{j,k}^{(o,\ell)}$ . Thus, we have

$$\Theta_j^{(o,\ell)} = \underset{\Theta_j^{(o)}}{\operatorname{argmin}} \sum_{m,n} \left( x_{i,j,k}(m, n) - (\mathbf{w}_{j,k}^{(o,\ell)})^\top \mathbf{y}_{j,k}^{(o)}(m, n; \Theta_j^{(o)}) \right)^2. \quad (11)$$

These two steps are alternatively repeated until the convergence of our algorithm. This alternating optimization technique is summarized in the following algorithm.

**Algorithm 1:** Alternating optimization

For each resolution  $j$  and subband  $o$ :

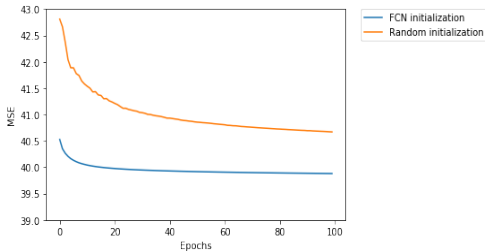
- ① Initialize the weights  $\Theta_j^{(o,0)}$
- ② for  $\ell = 1, 2, \dots$ 
  - (a) For each image  $k$ , compute  $\mathbf{y}_{j,k}^{(o)}$  using the weights  $\Theta_j^{(o,\ell-1)}$  and deduce the optimal weights of the output layer  $\mathbf{w}_{j,k}^{(o,\ell)}$  using Eq. (10).
  - (b) Update the parameters  $\Theta_j^{(o,\ell)}$  by minimizing the loss function defined in (11) using the MBGD algorithm.

It is worth noting that Step ②(a) is performed for each image of the training dataset. Thus, the mini-batch size (i.e. number of training samples used to compute the gradient and update the parameters of the model) correspond to the number of pixels to be predicted for each input image  $x_{j,k}$ . Moreover, in practice, the adaptive weights  $\mathbf{w}_{j,k}^{(o)}$  should be computed for each test image and sent to the decoder, which generates a very small transmission overhead.

#### 4. EXPERIMENTAL RESULTS

To evaluate the proposed FCN-based LS design methods, we have used the CLIC image dataset<sup>1</sup>. This dataset has been recently made available in the context of a workshop related to the Challenge on Learned Image Compression (CLIC 2018). For the training phase, the dataset is composed of 585 images with different sizes and the simulations are carried out by using Keras with TensorFlow backend on an NVIDIA Tesla V100 32 GB GPU. For the test phase, we have selected 35 images from the test CLIC dataset, which are then cropped into  $512 \times 512$  non-overlapping patches, yielding 55 test images. Note that this dataset is composed of color images which have been transformed to grayscale images coded at 8 bpp. Note that coding the luminance component is generally more challenging than the chrominance ones in color images.

The proposed methods are compared to the conventional 5/3 linear transform used in the lossless compression mode of JPEG2000. Note that this transform is referred to as Fix-NSLS since the weights of prediction and update filters are kept fixed. A comparison to its optimized version (OPT-NSLS) is also performed where prediction and update filters are optimized as described in Section 2. For a fair comparison, the proposed FCN-based lifting construction methods have been tested by considering the same supports of the prediction and update filters. The two FCN architectures are implemented using  $L = 3$  hidden layers with dimensions  $128 \times 64 \times 32$  neurons. Moreover, the Parametric Rectified Linear Unit (PReLU) has been employed as an activation function. As mentioned earlier, we used a dynamic mini-batch size corresponding to the number of pixels to be predicted in each image. For the proposed alternating optimization technique, we have considered 5 iterations to optimize the weight parameters for each input training image and 60 epochs for the convergence of the algorithm, while initializing the weights  $\Theta_{j,k}^{(o,0)}$  to the values obtained with the classical FCN model. Fig. 3 shows the convergence of the loss function of our proposed model during the first prediction stage of the NSLS using random and FCN initializations.



**Fig. 3.** Loss function evolution for the proposed alternating optimization technique.

Note that all these methods are carried out over three resolution level. For both the FCN- and DFCN-based methods, all the prediction and update models are separately trained at the different resolution levels. These methods are evaluated in terms of energy compaction of the detail wavelet coefficients using MSE and PSNR criteria. In addition, the entropy of the resulting wavelet representation is also computed in order to have an evaluation of the resulting bitrate in the context of lossless coding.

Table 1 shows the average prediction performance, evaluated in terms of MSE and PSNR computed over all the detail coefficients of the test images. The average entropy of the image wavelet representations is also provided. To this end, a rounding operator is applied

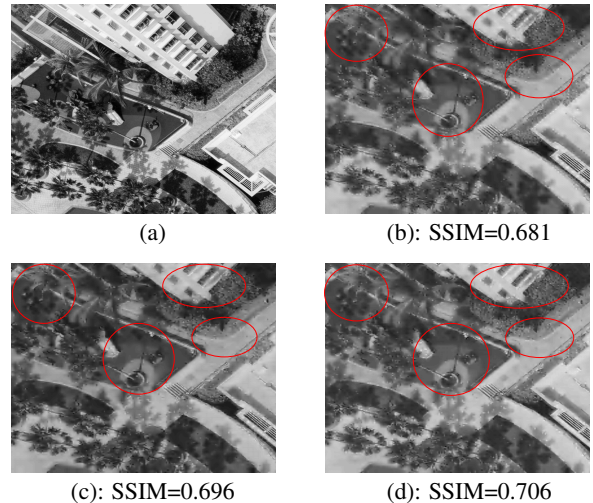
to the predicted value to generate integer wavelet coefficients. First, it can be noticed that the classical FCN-based model results in a gain of about 0.3-0.5 dB in terms of PSNR compared to the fixed as well as the optimized NSLS. Further improvements of about 0.75 dB in terms of PSNR are achieved by the proposed dynamic FCN-based model. Moreover, in terms of entropy, the proposed dynamic FCN-based model achieves an average gain of about 0.1 bpp compared to the conventional fixed and optimized NSLS as well as the classical FCN-based approach. For instance, it has been observed that the gain with respect to the linear transforms is greater than 0.1 bpp for 22 test images and reaches up to 0.3 bpp. In addition to the lossless compression performance, some preliminary tests have been carried out in the context of lossy compression. Fig. 4 illustrates an example of reconstructed images after JPEG2000 coding/decoding at 0.35 bpp. It can be seen that the proposed dynamic FCN approach leads to better visual quality and sharper edges. This is also confirmed by the provided SSIM values.

#### 5. CONCLUSION AND PERSPECTIVES

In this paper, we have presented two FCN models to optimize the prediction and update operators for lifting-based coding schemes. While the first one exploits a classical FCN architecture, the second one uses a novel dynamical model taking into account the input image. Experimental results show the benefits of the proposed FCN-based approaches for building more efficient lifting schemes. Ongoing research aims at increasing the supports of prediction and update steps. Moreover, the use of a rate-distortion loss function is currently investigated.

**Table 1.** Comparison of different LS design methods.

Method	MSE	PSNR (in dB)	entropy (in bpp)
Fix-NSLS	170.09	27.75	4.34
OPT-NSLS	160.93	27.95	4.33
FCN-NSLS	149.21	28.27	4.32
DFCN-NSLS	<b>127.46</b>	<b>29.02</b>	<b>4.24</b>



**Fig. 4.** (a) Original test image. Zoom on the reconstructed images at 0.35 bpp using: (b) OPT-NSLS, (c) FCN-NSLS, (d) DFCN-NSLS.

<sup>1</sup><http://www.compression.cc/2018/challenge/>

## 6. REFERENCES

- [1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, April 1992.
- [2] T. Guo, H. S. Mousavi, T. H. Vu, and V. Monga, "Deep wavelet prediction for image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, USA, July 2017, pp. 104–113.
- [3] J.-H. Jacobsen, A. W. M. Smeulders, and E. Oyallon, "i-RevNet: Deep invertible networks," in *International Conference on Learning Representations*, Vancouver, Canada, May 2018, pp. 1–11.
- [4] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, 1998.
- [5] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Norwell, MA, USA, 2002.
- [6] M. Kaaniche, A. Benazza-Benyahia, B. Pesquet-Popescu, and J.-C. Pesquet, "Vector lifting schemes for stereo image coding," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2463–2475, 2009.
- [7] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001, vol. 3, pp. 1793–1796.
- [8] E. Martinez-Enriquez, J. Cid-Sueiro, F. Diaz de María, and A. Ortega, "Directional transforms for video coding based on lifting on graphs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 4, pp. 933–946, November 2016.
- [9] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, no. 2, pp. 186–200, April 1996.
- [10] F. J. Hampson and J.-C. Pesquet, "M-band nonlinear subband decompositions with perfect reconstruction," *IEEE Transactions on Image Processing*, vol. 7, pp. 1547–1560, November 1998.
- [11] J. Solé and P. Salembier, "Generalized lifting prediction optimization applied to lossless image compression," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 695–698, October 2007.
- [12] Y. Liu and K. N. Ngan, "Weighted adaptive lifting-based wavelet transform for image coding," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 500–511, April 2008.
- [13] M. Kaaniche, A. Benazza-Benyahia, B. Pesquet-Popescu, and J.-C. Pesquet, "Non separable lifting scheme with adaptive update step for still and stereo image coding," *Elsevier Signal Processing: Special issue on Advances in Multirate Filter Bank Structures and Multiscale Representations*, vol. 91, no. 12, pp. 2767–2782, January 2011.
- [14] A. Gouze, M. Antonini, M. Barlaud, and B. Macq, "Design of signal-adapted multidimensional lifting schemes for lossy coding," *IEEE Transactions on Image Processing*, vol. 13, no. 12, pp. 1589–1603, December 2004.
- [15] M. Kaaniche, B. Pesquet-Popescu, A. Benazza-Benyahia, and J.-C. Pesquet, "Adaptive lifting scheme with sparse criteria for image coding," *EURASIP Journal on Advances in Signal Processing: Special Issue on New Image and Video Representations Based on Sparsity*, vol. 2012, no. 1, pp. 1–22, January 2012.
- [16] B. Pesquet-Popescu, *Two-stage adaptive filter bank*, first filling date 1999/07/27, official filling number 99401919.8, European patent number EP1119911, 1999.
- [17] R. Xiong H. Ma, D. Liu and F. Wu, "iWave: CNN-based wavelet-like transform for image compression," *accepted in IEEE Transactions on Multimedia*, December 2019.
- [18] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Computer Vision and Pattern Recognition*, Las Vegas, USA, June 2016, pp. 5306–5314.
- [19] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *International Conference on Machine Learning*, Sydney, Australia, August 2017, pp. 1–9.
- [20] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, Toulon, France, April 2017, pp. 1–27.
- [21] M. Kaaniche, J.-C. Pesquet, A. Benazza-Benyahia, and B. Pesquet-Popescu, "Two-dimensional non separable adaptive lifting scheme for still and stereo image coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, Texas, USA, March 2010.
- [22] Y.-K. Sun, "A two-dimensional lifting scheme of integer wavelet transform for lossless image compression," in *International Conference on Image Processing*, Singapore, October 2004, vol. 1, pp. 497–500.
- [23] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.
- [24] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, San Diego, USA, May 2015, pp. 1–15.